Sustainable Structure and Materials, *Vol. 3, No .1, (2020) 25-36*
DOI: https://doi.org/10.26392/SSM.2020.03.01.025

Prediction of Compressive Strength in High Strength Concrete with Steel Fiber Addition using Support Vector Machine Algorithm

Abdulhameed Umar Abubakar[1]*, Kachalla Alhaji Kau[1], Murtala Hassan[1], Maimuna Salisu Tabra [2]

[1]Civil Engineering Department, Modibbo Adama University of Technology, Yola, Nigeria
[2]Department of Computer Science, Gombe State University (GSU), Gombe, Nigeria

*Corresponding E-mail: abdulhameed.umar@mautech.edu.ng

(*Received April 04, 2020, Revised May 28, 2020, Accepted June 02, 2020*)

ABSTRACT. *In this study, a support vector machine model available in Weka Algorithms, was utilized to test the predictive capacity of compressive strength in high strength concrete (HSC) with steel fiber addition. To test the performance of the algorithm, a certain percentage were allocated for training of the algorithm, and the rest for test. This was done from 60-40 percent split up to 90-10 percent split for training and testing respectively. Results generated from the model include mean absolute error(MAE), root mean squared error (RMSE), and relative absolute error (RAE) for each model. It was observed that the correlation coefficient for all the percent split was 0.82, and the highest and lowest MAE were 9.969 and 9.4714 respectively, an indication of reliability and precision. Utilization of free algorithms in civil engineering construction will enhance the optimization of concrete mixtures.*

Keywords: High strength concrete, steel fiber-reinforced concrete, compressive strength prediction, algorithms, support vector machine.

1. INTRODUCTION

Support Vector Machines (SVM) are systems which uses hypothesis space of a linear function in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory (SLT). They belong to a family of generalized linear classifiers. In other words, SVM is a classification and regression prediction tool that uses machine learning theory (MLT) to maximize predictive accuracy while automatically avoiding over-fit to the data [1]. SVMs are a set of related supervised learning methods used for classification and regression [2].

SVM has found wide application in face recognition software, time series prediction, and medical diagnosis, [3 - 5]. These successes have further invigorated research to widen their applications [6]. Support vectors are data points that lie closest to the decision surface or hyperplanes [7]. It is extremely difficult to classify these data points because they have direct bearing on the optimum location of the decision surface to find an optimal solution. This is done by maximizing the margin around the separating hyperplane.

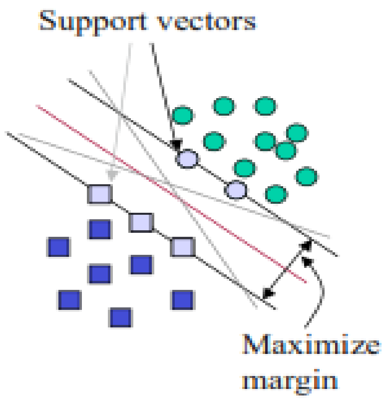

Fig.-1: Support Vectors around a hyperplane [7]

The simplest formulation of SVM is linear where the hyperplane lies on the space of the input data x:

$$f(x) = w.x + b \tag{1}$$

Where:
w = weight vector
x = input vector
b = bias

SVM finds a hyperplane in a space different from that of the input data x. It is a hyperplane in a feature space induced by a Kernel (the Kernel defines a dot product in that space) [8]. The concept of a Kernel was explained in [1] as "if a data is linear, a separating hyperplane may be used to divide the data. In cases where the data is nonlinear, Kernel are used to map the input data to a high-dimensional space, thus making the new data separable [2]". This introduces a new concept, the 'Kernel Function' which enable operations to be performed in the input space rather than potentially high-dimensional feature space. Through the Kernel, the hypothesis space is defined as a set of "hyperplanes" in the feature space induced by K. This can be seen as a set of functions in a Reproducing Kernel Hilbert Space (RKHS) defined by K [8-9]. Further discussion on Kernel function in SVM and its performance can be found elsewhere [10].

2. GENERAL INPUTS/OUTPUTS IN SVM

In SVM, the set of training pair samples (input, output) are X_1, $X_2 \ldots X_n$, as the inputs while the output result is y. A set of weights w (or w_i) one for each feature, whose linear combination predicts the value of y, as what is obtainable in neural networks. However, the significant difference is the use of optimization of maximizing the margin (street width – hyperplane) to reduce the number of weights that are nonzero to just a few that correspond to the important features that plays a role in deciding the separating line (hyperplane). These nonzero weights correspond to the support vectors as seen in Fig. 2, because they "support" the separating hyperplane [6].

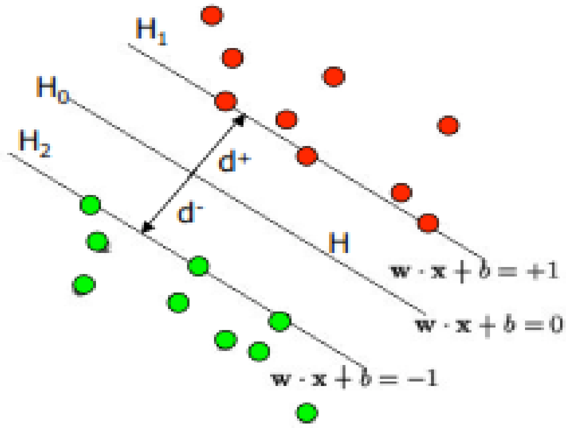

Fig.-2: Support Vectors ‘supporting’ a separating hyperplane [7]

Input vectors that just touch the boundary of the margin are defined as in Fig. 2 where:

H = hyperplanes
d = shortest distance to the closet point.

The key advantages of neural network algorithms are that it has the ability to learn, recognize, generalize, classify, and interpret incomplete and noisy inputs to represent both linear and nonlinear relationships with great accuracy [11]. Artificial neural networks (ANN) have been successfully used to predict multiple variables and nonlinear behavior of different parameters in the concrete mixture to obtain compressive strength under different ages [12-16].

To minimize the experimental task of concrete mix design, probabilistic models are generally constructed and constitutive equations are derived [17]. Regression analysis though quicker and simpler in making predictions, the accuracy is found to reduce as the number of independent variables increases [18]. Therefore, in this kind of situations, the use of algorithm related programs is more accurate to predict the models. ANNs have also been used to optimize the proportion of four concrete ingredients (water, cement, fine, and coarse aggregates) [19] were used to predict 28-day compressive strength of HPC with six components (cement, silica fume, superplasticizer, water, fine and coarse aggregates) has been predicted using fuzzy-ARTMAP ANNs [20]. A combination of fuzzy neural networks and polynomial neural networks has been experimented by [21] with six input parameters (ingredients) and 28-day compressive strength as the output parameter.

The originality of this study is aimed at utilizing free algorithms that were otherwise used for classification and clustering of data in computer science to be able to predict compressive strength of HSC with steel fiber addition. The significant contribution of this paper is in customizing the use of this algorithms and their potential applicability in civil engineering. Also, worth mentioning is the use of ten (10) attributes were previous studies using ANN [19 – 20] have been limited to six. The study is limited to SVM alone using data collected from literature.

3. SOFTWARE ARCHITECTURE

The algorithm used is Sequential Minimal Optimization (SMO) credited to John Platt [22] that has been customized in Weka Algorithm. It provides an efficient way of solving dual problem arising from the derivation of SVM. It has been widely applied to pattern classification problems and nonlinear regressions [23]. The SVM is trained as a classifier using a portion of the training dataset, then use the classifier that was trained to classify the remaining data. They find an optimal solution to data points that are more difficult to classify.

3. PREDICTION METHODOLOGY

The software utilized was Weka Version 3.8.3, an open source Java based machine learning algorithm created by University of Waikato, New Zealand. To utilize the software, the inputs known as ‘attributes’ had to be prepared in either csv or arff format as seen in Fig. 3 where all the attributes (parameters) were defined, including the response compressive strength. It should be noted that the attributes are written as a single word for titles with more than one words such as “FiberDiameter” or “FiberLength”. The data used in this study was obtain from sources in Table 1 and entered as according to how the attributes were arranged in Table 2. If a mistake is made during entry without following the pre-defined format, the program will return an error message specifying the line number where the problem is located. All the data for the attributes were placed, each separated by a ‘comma’ and saved.

Prediction - Notepad

File Edit Format View Help

```
@relation Prediction

@attribute cement real
@attribute water real
@attribute coarse real
@attribute fine real
@attribute silicaFume real
@attribute HRWR real
@attribute steelFiber real
@attribute compressiveStrength real

@data
470,165,1050,700,47,14,0,86
470,165,1050,700,47,14,0.5,88.97
470,165,1050,700,47,14,0.5,89.03
470,165,1050,700,47,14,0.5,89.15
470,165,1050,700,47,14,0.5,89.23
470,165,1050,700,47,14,0.5,89.99
470,165,1050,700,47,14,0.5,90.06
470,165,1050,700,47,14,0.5,90.16
470,165,1050,700,47,14,0.5,90.21
470,165,1050,700,47,14,0.5,90.95
```

Fig-3: Example of an arff or csv file created using Notepad used in preprocessing

In the Weka Software interface, the menu “Explorer” is selected followed by “Preprocess” where the csv or arff file is uploaded. Next, “Classify” input function is selected followed by “Percentage Split”. This is because a certain portion of the data would be used for training, and the rest for test. In here, 60-40 % was the initial starting point for training-testing. This was continued with an increment of 10 % up to 90-10 % for training and testing. This is to ascertain the influence of percent split on the algorithm. The “Choose” function on the software interface, followed by SMO was selected. After running the program, the output is displayed in the right hand corner of the software interface, however, the programmer has to decide the mode of storage. It can be stored in CSV format in the form of excel spreadsheet, plain text that can be opened with text editors, or store the model for future use when more data is available.

Table-1: Sources of Data from Literature

S/No	References
1.	Ackgenc *et al.*, [24]
2.	Abubakar [25]
3.	Eren & Marar [26]
4.	Eren *et al.*, [27]
5.	Ibrahim & Che Bakar [28]
6.	Marar *et al.*, [29]
7.	Nguyen-Minh *et al.*, [30]
8.	Nili & Afroughsabet [31]
9.	Pigeon & Cantin [32]
10.	Sahin & Koksal [33]
11.	Unal *et al.*, [34]
12.	Yalcin [35]

Table-2: Attributes with maximum and minimum values used

Cement (kg/m^3)	**Water** (kg/m^3)	**Dmax** (mm)	**Coarse** (kg/m^3)	**Fine** (kg/m^3)	**SP** (kg/m^3)	V_f (%)	L_f (mm)	D_f (mm)	f_c (MPa)
565-288	230.2-123	31.5-10	1398.8-749.2	1064.2-530	17-0.5	2-0.19	60-30	1-0.1	116-18.18

SP: Superplasticizer, V_f: Fiber Volume, L_f: Fiber Length, D_f: Fiber Diameter

4. RESULTS

An annotated computer printout of the result (see Appendix) for SVM - SMO algorithm is presented for the training-to-testing percentage of 90 - 10. The total number of datasets called "instances" clearly stated as well as the attributes (parameters), followed by the percentage of split; and the time taken to generate the model. This is followed by four columns of data (serial number, actual dataset, predicted dataset, and the error), each significant.

The left hand column gives the actual data that was used for testing, and the right hand side, the prediction output. At the extreme right hand column, the prediction error result is presented for the instances. In the error, it could be seen that some data have negative signs attached to them while others are in the positive territory with a very high values. The former shows an underestimation of the values, while the latter indicates an overestimation, sometimes gross over or under estimation by the model occur. Values that are very close zero indicates they are closer to the actual value because the prediction error is small, values that equals zero shows the prediction efficiency.

The printout concludes by presenting a summary of model performance evaluation depicted in Table 3. First is Correlation Coefficient (CC) which measures the statistical correlation between the predicted values and the actual values, for good performance of the model, large values should be expected. This should not be confused with coefficient of determination (R^2) which measures the quality describing the proportion of variability explained by the fitted model.

Table-3: Model Evaluation

Training-to-Testing Split	60	70	80	90
Correlation Coefficient	0.8247	0.8198	0.819	0.8242
Mean Absolute Error	9.7865	9.5017	9.4714	9.969
Root Mean Squared Error	13.1872	13.2857	12.9453	13.9515
Relative Absolute Error %	49.8779	48.5766	50.4762	47.8588
Root Relative Sq. Error %	57.335	57.9612	57.3818	56.6328
Instances	77	58	38	19

Mean Absolute Error (MAE) is another index that measures the performance of a model. It measures how close predictions are to the eventual outcomes by measuring average magnitude of the errors in a set of forecasts:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| \quad (2)$$

Where f_i = prediction; y_i = true value

To measure the differences between values (sample values) predicted by a model and the values actually observed, Root Mean Square Error (RMSE) was also utilized. This represents the sample standard deviation of the differences between predicted values and observed values.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(f_i - y_i)^2}{n}} \quad (3)$$

RMSE is preferred when large errors are undesirable, and it is always larger than MAE.

Two other errors evaluated were Root Absolute Error (RAE) and Root Relative Squared Error (RRSE) where the former shows much the results deviates from actual value, and the latter is a measure in percent compared to the actual value. Both shows how far the prediction deviates from the actual value.

$$RAE = \frac{\sum_{i=1}^{n}|f_i - y_i|}{\sum_{i=1}^{n}|t_i - y_i|} \quad (4)$$

Where t_i = mean value of y

$$RRSE = \sqrt{\frac{\sum_{i=1}^{n}(f_i - y_i)^2}{\sum_{i=1}^{n}(t_i - y_i)^2}} \quad (5)$$

Lower values of these errors and a high CC values indicates that the model is suitable and prediction accuracy is very high, however, if the results is otherwise, it is an indication that more data might be required to improve the model performance.

Results generated by the algorithm were plotted against the actual results and presented in Fig. 4 – 7. Coefficient of determination R^2 was also determined after curve fitting, where quadratic models were found to be the best fit.

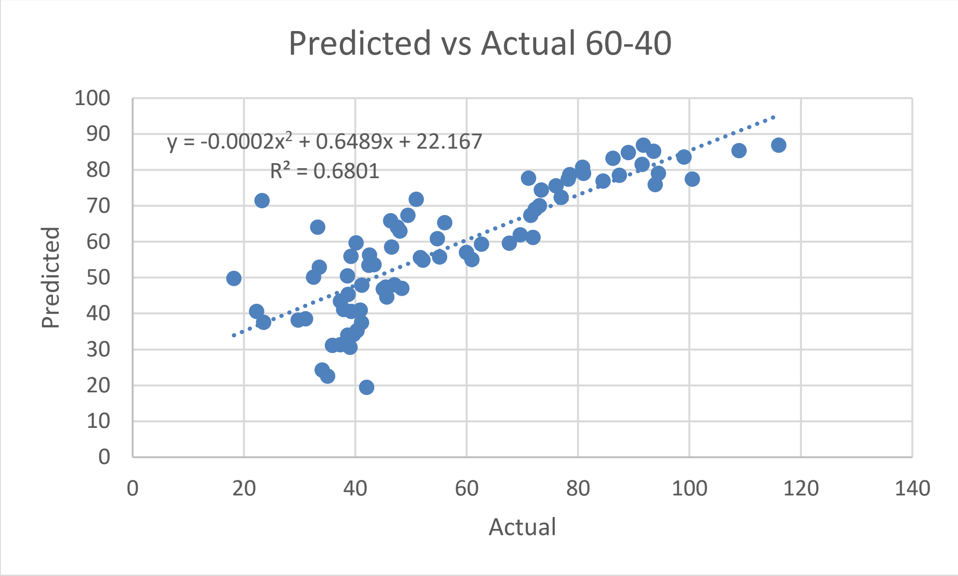

Fig-4: Prediction efficiency for 60 – 40 split

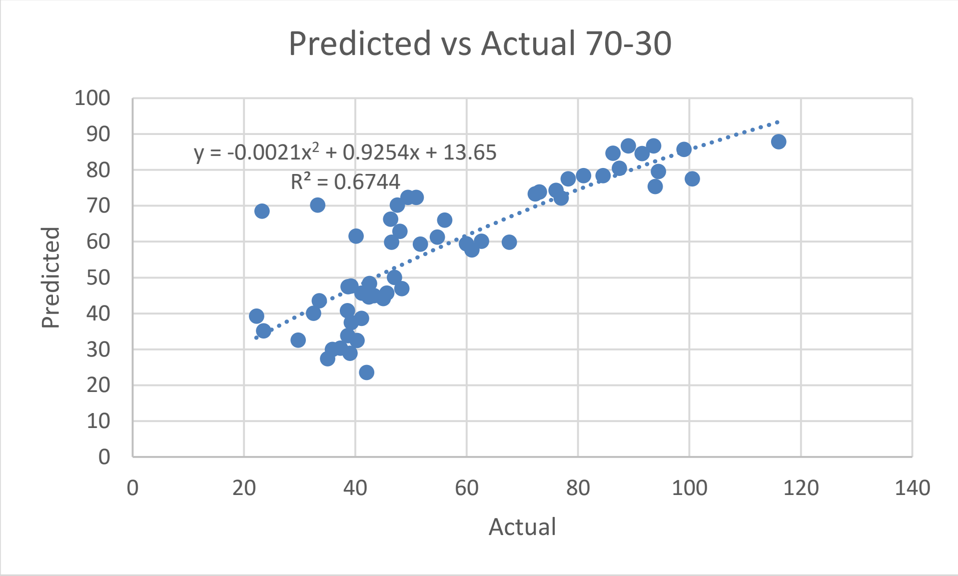

Fig-5: Prediction efficiency for 70 – 30 split

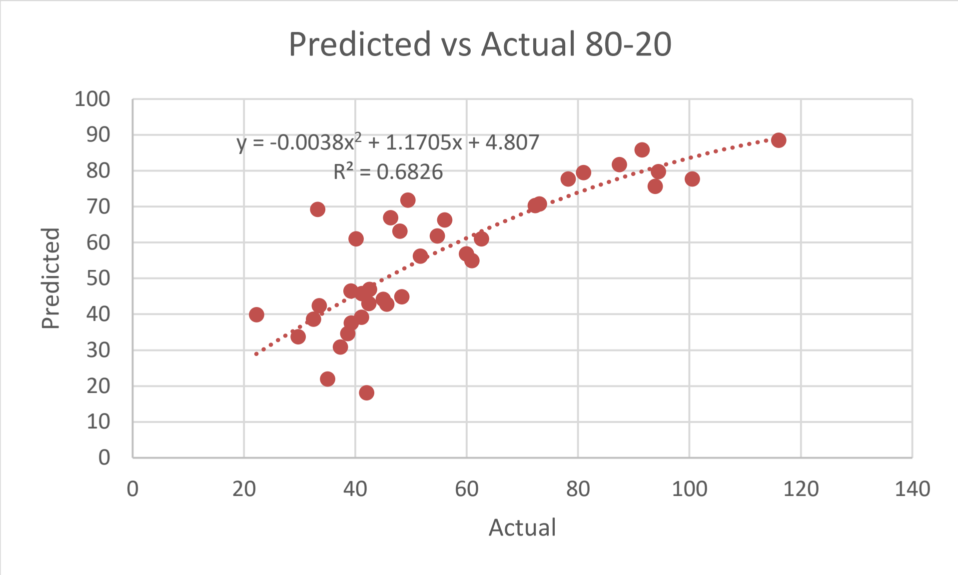

Fig-6: Prediction efficiency for 80 – 20 split

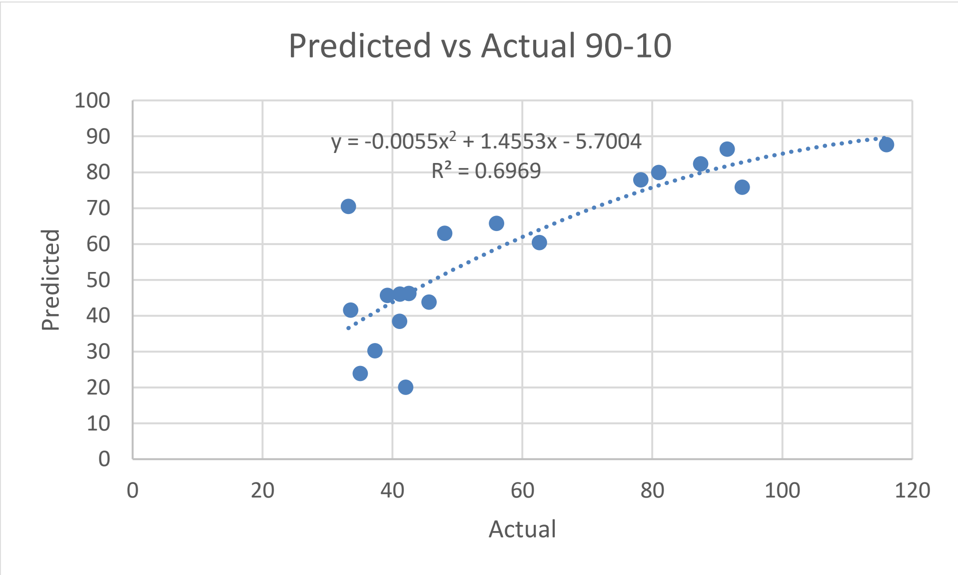

Fig-7: Prediction efficiency for 90 – 10 split

5. CONCLUSIONS

This study involving HSC with steel fiber addition evaluated the performance of SMO – SVM in Weka Software in the prediction of concrete compressive strength and the following conclusions have been drawn:

- WEKA software was able to be trained to utilize attributes (concrete ingredients) to give an appropriate output that can be replicated.
- To validate the performance of the models, correlation coefficient (CC) measured were 82 % for all the percentage splits, and it is seen that increasing the percentage of training dataset up to 90% did not have significant effect on the CC values.

- Quadratic model was the best fit with R^2 of 0.68, 0.67, 0.68 & 0.69 for the percent splits respectively.

6. RECOMMENDATIONS

Future study should explore the possibility of increasing the sample size, as well as other methods such as Bootstrap and Cross Validation.

REFERENCES

[1] Jakkula, V. (N.D.) "Tutorial on Support Vector Machines" available online at course.ccs.edu./cs5100f11/resources/jakkula.pdf

[2] Wikipedia Online. http://en.wikipedia.org/wiki

[3] Tefas, A., Kotropoulous, C., Pitas, I. (1999) "Enhancing the performance of elastic graph matching for face authentications by using support vector machines". In Proceedings of Advanced Course on Artificial Intelligence (ACAI 99), Chania, Greece.

[4] Fernandez, R. (1999) "Predicting time series with a local support vector regression machine". In Proceedings of Advanced Course on Artificial Intelligence (ACAI 99), Chania, Greece.

[5] Veropoulos, K., Cristianni, N., Campbell, C. (1999) "The Application of Support Vector Machines to Medical Decision Support: A Case Study". In Proceedings of Advanced Course on Artificial Intelligence (ACAI 99), Chania, Greece.

[6] Evgeniou, T., Pontil, M. (1999) "Workshop on Support Vector Machines: Theory and Applications". In Proceedings of Advanced Course on Artificial Intelligence (ACAI 99), Chania, Greece. DOI:10.1007/3-540-44673-7_12

[7] Berwick, R. (N.D.) "An Idiot's Guide to Support Vector Machines (SVM)" available at web.mit.edu/6.034/wwwbob/sum-notes-long-09.pdf

[8] Wahba, G. (1990) "Splines Models for Observational Data". Series in Applied Mathematics, Vol. 59, SIAM.

[9] Vapnik, V. (1998) "Statistical Learning Theory". Wiley, New York.

[10] Cristianni, N., Shawe-Taylor, J. (2000) "An Introduction to SVM and Other Kernel-based Learning Methods". Cambridge University Press.

[11] Faruqi, M.A., Agarwala, R., Sai, J., Francisco, A. (2015) "Application of Artificial Intelligence to Predict Compressive Strength of Concrete from Mix Design Parameters: A Structural Engineering Application", *Journal of Civil Engineering Research*, 5(6), 158-161.

[12] Chou, J.-S. C., Tsai, C.-F. (2012) "Concrete compressive strength analysis using a combined classification and regression technique", *Automation in Construction*, 24, 52-60.

[13] Hong-Guang, N., Wang, J.-Z. (2000) "Prediction of compressive strength of concrete by neural networks", *Cement and Concrete Research*, 30 (8), 1245-1250.

[14] Saridemir, M. (2009a) "Prediction of compressive strength of concretes containing metakaolin and silica fume by artificial neural networks", *Advances in Engineering Software*, 40 (5), 350-355.

[15] Saridemir, M. (2009b) "Predicting the compressive strength of mortars containing metakaolin by artificial neural networks and fuzzy logic", *Advances in Engineering Software*, 40 (9), 920-927.

[16] Duan, Z. H., Kou, S. C., Poon, C. S. (2012) "Prediction of compressive strength of recycled aggregate concrete using artificial neural networks", *Construction and Building Materials.*

[17] Chithra, S., Senthil Kumar, S.R.R., Chinnaraju, K., Ashmita, F.A. (2016) "A Comparative Study on the Compressive Strength prediction models for High Performance Concrete containing nano silica and copper slag using regression analysis and Artificial Neural Networks", *Constr. Build. Mater.*, 114, 528-535.

[18] Sadrmomtazi, A., Sobhani, J., Mirgozar, M.A. (2013) "Modeling compressive strength of EPS lightweight concrete", *Constr. Build. Mater.*, 42, 205-216.

[19] Oh, J., Lee, I., Kim, J., & Lee, G. (1999). Applications of neural networks for proportioning of concrete mixes. *ACI Material Journal*, 96 (1), 51–59.

[20] Kasperkiewicz, J., Dubrawski, A. (1995) "HPC strength prediction using artificial neural network", *Journal of Computing in Civil Engineering*, 9 (4), 279–284.

[21] Fazel Zarandi, M.H., Türksen, I.B., Sobhani, J., Ramezanianpour, A.A. (2008) "Fuzzy polynomial neural networks for approximation of the compressive strength of concrete", *Applied Soft Computing* 8 (1), 488–498.
[22] Platt, J.C. (1998). Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Available at www.microsoft.com MSR-TR-98-14
[23] Chu, F., Jin, G., Wang, L. (2005) "Cancer Diagnosis and Protein Secondary Structure Prediction using Support Vector Machines", *StudFuzz*, Vol. 177, pp. 343-363.
[24] Ackgenc, M., Ulas, M. & Alyamac, K. E. (2015). Using Artificial Neural Network to Predict Mix Compositions of Steel Fiber – Reinforced Concrete. *Arab J. Sci. Eng. Vol. 40*, 407 – 419.
[25] Abubakar, A.U. (2018) "Influence of Steel Fiber Addition on Workability & Mechanical Behavior of High Performance Concrete" PhD Thesis, EMU North Cyprus.
[26] Eren, O. & Marar, K. (2009). Effects of limestone crusher dust and steel fibers on concrete. *Constr. Build. Mater.*, 23, 981-988. doi:10.1016/j.conbuildmat.2008.05.014
[27] Eren, O., Marar, K. & Celik, T. (1999). Effects of Silica Fume and Steel Fibers on Some Mechanical Properties of High-Strength Fiber – Reinforced Concrete. *Journal of Testing & Evaluations, JTEVA Vol. 27, No. 6*, 380 – 387.
[28] Ibrahim, I.S., & Che Bakar, M.B. (2011). Effects on mechanical properties of industrialised steel fibers addition to normal weight concrete. In: Proceedings of The Twelfth East Asia-Pacific Conference on Structural Engineering and Construction. Procedia Engineering. doi:10.1016/j.proeng.2011.07.329
[30] Marar, K., Eren, O. & Yitmen, I. (2011). Compression Specific Toughness of Normal Strength Steel Fiber Reinforced Concrete (NSSFRC) and High Strength Steel Fiber Reinforced Concrete (HSSFRC). *Materials Research, 14(2)*, 239-247.
[31] Nguyen-Minh, L., Rovnak, M., Tran-Quoc, T., & Nguyenkim, K. (2011). Punching Shear Resistance of Steel Fiber Reinforced Concrete Flat Slabs. In: Proceedings of The Twelfth East Asia-Pacific Conference on Structural Engineering and Construction. Procedia Engineering. doi:10.1016/j.proeng.2011.07.230
[32] Nili, M. & Afroughsabet, V. (2010). Combined effect of silica fume and steel fibers on the impact resistance and mechanical properties of concrete. *Int. J. Impact Eng.*, 32, 879-886. doi:10.1016/j.ijimpeng.2010.03.004
[33] Pigeon, M. & Cantin, R. (1998). Flexural properties of steel fiber-reinforced concretes at low temperatures. *Cem. Concr. Compos.*, 20, 365-375. doi:10.1016/S0958-9465(98)00017-1
[34] Sahin, Y. & Koksal, F.(2011). The influences of matrix and steel fiber tensile strengths on the fracture energy of high-strength concrete. *Constr. Build. Mater.*, 25, 1801-1806. doi:10.1016/j.conbuildmat.2010.11.084
[35] Unal, O., Demir, F. & Uygunoglu, T. (2007). Fuzzy logic approach to predict stress–strain curves of steel fiber-reinforced concretes in compression. *Build. Environ.* doi:10.1016/j.buildenv.2006.10.023
[36] Yalcin, M.(1994). Optimization and Performance Based Design of Steel Fiber Reinforced Concretes. Doctoral thesis, Istanbul Technic University, Civil Engineering Faculty, Turkey.

APPENDIX

A Typical Weka Software printout for SMO 90-10

```
=== Run information ===
Scheme:     weka.classifiers.functions.SMOreg -C 1.0 -N 0 -I "weka.classifiers.functions.supportVector.
Relation:   Prediction
Instances:  192
Attributes: 10
            cement
            water
            Dmax
            coarse
            fine
            superplasticizer
            volumefraction
            fiberlength
            fiberdiameter
            compressiveStrength
Test mode:  split 90.0% train
 === Classifier model (full training set) ===
 SMOreg
 weights (not support vectors):
  +   0.2088 * (normalized) cement
  -   0.1966 * (normalized) water
  -   0.2473 * (normalized) Dmax
  -   0.1409 * (normalized) coarse
  -   0.4128 * (normalized) fine
  -   0.0754 * (normalized) superplasticizer
  +   0.1501 * (normalized) volumefraction
  +   0.0159 * (normalized) fiberlength
  -   0.088  * (normalized) fiberdiameter
      0.798
 Number of kernel evaluations: 18528 (95.68% cached)
 Time taken to build model: 0.04 seconds
 === Classifier model for training split (173 instances) ===
 SMOreg
 weights (not support vectors):
  +   0.2166 * (normalized) cement
  -   0.3157 * (normalized) water
  -   0.2643 * (normalized) Dmax
  -   0.1081 * (normalized) coarse
  -   0.5398 * (normalized) fine
  -   0.136  * (normalized) superplasticizer
  +   0.1623 * (normalized) volumefraction
  +   0.0187 * (normalized) fiberlength
  -   0.1113 * (normalized) fiberdiameter
      0.9672
```

Number of kernel evaluations: 15051 (94.938% cached)

=== Predictions on test split ===

inst#	actual	predicted	error
1	80.95	79.985	-0.965
2	56	65.792	9.792
3	42	20.12	-21.88
4	116	87.694	-28.306
5	41.13	46.068	4.938
6	41.08	38.437	-2.643
7	33.52	41.605	8.085
8	42.51	46.258	3.748
9	78.2	77.894	-0.306
10	91.49	86.442	-5.048
11	93.8	75.861	-17.939
12	33.2	70.517	37.317
13	87.4	82.375	-5.025
14	35	23.926	-11.074
15	39.2	45.672	6.472
16	48	62.989	14.989
17	45.6	43.842	-1.758
18	62.6	60.441	-2.159
19	37.27	30.303	-6.967

=== Evaluation on test split ===

Time taken to test model on test split: 0.02 seconds

=== Summary ===

Correlation coefficient	0.8242
Mean absolute error	9.969
Root mean squared error	13.9515
Relative absolute error	47.8588 %
Root relative squared error	56.6328 %
Total Number of Instances	19